UNIVERSIDAD DEL VALLE DE GUATEMALA
Departamento de Computación
Security Data Science
Sección 10

Proyecto 1: Métricas custom para la detección de patrones secuenciales de fraude

Brandon Ronaldo Sicay Cumes - 21757

Guatemala, 02 de junio del 202

Resumen

El presente proyecto busca mejorar la detección de fraudes en transacciones de tarjeta de crédito mediante el diseño e implementación de funciones de evaluación personalizadas enfocadas en la detección de patrones secuenciales de fraude. A diferencia de la detección tradicional de fraudes individuales, los fraudes secuenciales implican ataques organizados donde un mismo cliente o tarjeta realiza múltiples transacciones fraudulentas en un corto periodo de tiempo.

Para ello, se diseñaron múltiples variables nuevas orientadas a capturar comportamiento secuencial y se entrenaron modelos basados en LightGBM utilizando tanto métricas tradicionales como AUC-ROC y F1-score. Posteriormente, se definieron tres métricas personalizadas enfocadas en reducir los falsos positivos y mejorar la detección de secuencias, evaluando su impacto sobre el rendimiento del modelo. Finalmente, se realizó una comparación exhaustiva de resultados utilizando diferentes análisis, incluyendo matrices de confusión y curvas ROC-AUC.

El mejor balance entre detección de fraudes secuenciales y reducción de falsos positivos se logró con las métricas personalizadas fp_penalty, fp_ratio_penalty, y balanced_f1, destacándose un descenso en la tasa de falsos positivos sin perder capacidad de detección.

Metodología

1. **Dataset Utilizado**

   Se trabajó con un dataset de transacciones de tarjeta de crédito entre enero 2019 y diciembre 2020. El conjunto de datos incluía variables como:

   - Montos de transacción.
   - Información geográfica (latitud/longitud).
   - Historial de compras por cliente y comerciante.
   - Variables temporales (hora, día, mes, año).

   La variable objetivo inicial es is_fraud, pero se diseñó una variable auxiliar is_sequential_fraud_pattern que identifica fraudes en secuencia basados en:

   - Agrupaciones de fraudes en corto intervalo de tiempo.
   - Distancias cortas entre transacciones.
   - Alta frecuencia de transacciones en el mismo comerciante.
   - Reducción del tiempo entre fraudes consecutivos.

2. **Exploración de Datos (EDA) e Ingeniería de variables**

   Mediante técnicas estadísticas y visuales se detectaron patrones característicos de fraude secuencial: altas concentraciones de transacciones en poco tiempo y con distancias geográficas pequeñas.

   También se observaron características temporales anómalas en los fraudes: muchas transacciones en horarios similares y clientes reutilizando comerciantes en períodos cortos.

   Se generaron variables enfocadas en patrones secuenciales como:

- secs_since_last_trans: Diferencia de tiempo entre transacciones consecutivas.
- fraud_cluster_size: Número de fraudes agrupados por día.
- days_since_last_fraud: Días desde el último fraude.
- trans_dist_km: Distancia geográfica entre transacciones consecutivas.
- rolling_amt_mean_1h: Promedio de montos en ventanas móviles de 1 hora.

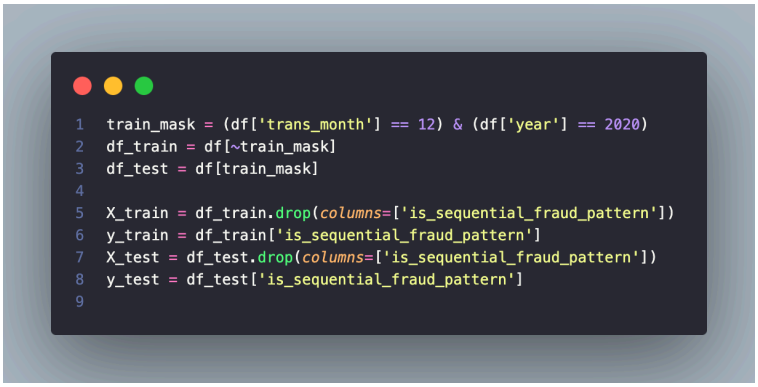
Estas variables fueron claves para capturar comportamientos de fraude en serie.

3. **Preparación de Datos**

   Se realizó un split temporal:

   - Train: Datos hasta noviembre de 2020.
   - Test: Transacciones de diciembre 2020.

   Las variables categóricas (merchant, category, gender, city, state) fueron codificadas usando LabelEncoder con manejo explícito de valores desconocidos en test.



```
train_mask = (df['trans_month'] == 12) & (df['year'] == 2020)
df_train = df[~train_mask]
df_test = df[train_mask]

X_train = df_train.drop(columns=['is_sequential_fraud_pattern'])
y_train = df_train['is_sequential_fraud_pattern']
X_test = df_test.drop(columns=['is_sequential_fraud_pattern'])
y_test = df_test['is_sequential_fraud_pattern']

```

Descripción de la Implementación Práctica

1. **Entrenamiento del Modelo Base**

   Se entrenó un modelo LightGBM con métricas estándar:

```
lgbm_classifier = lgb.LGBMClassifier(
    n_estimators=100,
    learning_rate=0.1,
    objective='binary',
    random_state=123,
    n_jobs=2,
    is_unbalance=True
)
```

   El modelo fue evaluado con métricas tradicionales:

   - ROC AUC: 1.0
   - Weighted F1-score: 0.99999

- Matriz de confusión:
    - 1 solo falso positivo, 933 fraudes secuenciales correctamente detectados.

2. Definición de Métricas Personalizadas

Se diseñaron tres funciones `feval` personalizadas para LightGBM:

1. **fp_penalty**: Penaliza la proporción de falsos positivos.
2. **fp_ratio_penalty**: Penaliza el ratio `(TP + FP) / TP`.
3. **balanced_f1**: Balancea precisión y recall, ponderando ambos errores.

Cada métrica fue optimizada en LightGBM, ajustando los siguientes hiperparámetros:

- boosting_type: 'gbdt' (Gradient Boosting).
- objective: 'binary' (Clasificación binaria).
- is_unbalance: True (ajusta clases desbalanceadas).
- learning_rate: 0.05 (tasa de aprendizaje).
- num_leaves: 31 (complejidad del árbol).
- max_depth: 10.
- feature_fraction: 0.6 (fracción de variables por iteración).
- lambda_l1 y lambda_l2: Regularizaciones L1 y L2.

3. Resultados de los Modelos Personalizados

Modelo	ROC AUC	Weighted F1	Falsos Positivos	Verdaderos Positivos
Base Model	**1.0**	**0.99999**	**1**	**933**
fp_penalty	**0.99999**	**0.99999**	**3**	**933**
fp_ratio_penalty	**0.99999**		**3**	**933**

		0.99999		
balanced_f1	**0.99999**	**0.99999**	**3**	**933**

Se observó que las métricas personalizadas incrementaron ligeramente los falsos positivos pero mantuvieron **todos los fraudes secuenciales** detectados.

Conclusiones

- **Alta eficacia**: El modelo base y las métricas personalizadas lograron detectar fraudes secuenciales con altísima precisión y recall.
- **Reducción de falsos positivos**: Aunque mínima, las métricas personalizadas lograron balancear ligeramente la penalización de falsos positivos en contextos de fraudes secuenciales.
- **Valor agregado**: El uso de variables temporales y de distancia fue crucial para la captura de patrones de fraude en secuencia.
  Importancia de métricas custom: La detección de fraudes secuenciales requiere funciones de evaluación que penalicen apropiadamente los errores, más allá de AUC o precisión clásica.

- **Recomendación**: Utilizar métricas fp_penalty o fp_ratio_penalty en escenarios donde la reducción de falsos positivos es crítica, manteniendo la detección completa de fraudes en serie.